custom module is a Custom Script Dynamically Linked Library (DLL) (122). A DLL is a feature of the Microsoft Windows® operating system and OS/2 that allows executable routines to be stored as files with DLL extensions and to be loaded only when needed by a program. In the present invention, the Custom Script DLL implements integration between a Security Management System with a client-side cryptographic function (128), *e.g.*, Entrust/Entelligence, and a PKI-Bridge (124) using a SmartDial Authentication Protocol (SDAP). The first and second custom modules are considered part of the client computer (102). A third custom element is the PKI-Bridge (124), which is a library and is stored on the server (112).

[0040] In addition to the custom modules, in one or more embodiments SmartDial may also contain a number of programming interfaces, such as an Application Programming Interface (API). The API is a set of routines used by an application program to direct the performance of procedures by the computer operating system. A first programming interface is a TCP/IP and Microsoft® RAS library (126) to allow the client computer (102) to connect to the server (112) through the PC modem (108) and the Remote Access Switch (110). A second programming interface is a client-side cryptographic function (128) allowing the dial-up client (120) and the Custom Script DLL (122) to connect to the card reader (104) for integration with a security device (106). Additionally, Schlumberger (SLB) proprietary Middleware (123), which is a library together with a proprietary smart card interface, *e.g.*, Microsoft® PCSC (127), and a proprietary smart card device driver, *e.g.*, Microsoft® SC drivers (129), facilitate the connection between the Custom Script DLL (122) and the card reader (104). A fourth programming interface is a server-side cryptographic function (130) to allow the server (112) and a Steel Belted RADIUS library (131) to integrate with a directory service (113) on the directory server (114).

[0041] Several of the main components of SmartDial listed above are described in greater detail below. Following the discussion of the main components is a description of a typical implementation of the components of SmartDial.

[0042] The dial-up client (120) is an executable file that loads and executes the code in the Custom Script DLL (122). Further, in one or more embodiments, there are two primary components within the dial-up client (120) that provide the necessary functionality to the dial-up client (120): a SDLogin component and a SDSetupDial component.

[0043] The SDLogin component is called by the dial-up client (120) when it is initially started, *i.e.,* the user double clicks an icon to start the application. The SDLogin component may be called again when dialing is about to begin. The SDLogin component allows the dial-up client (120) to logon onto the client-side cryptographic function (128).

[0044] The SDSetupDial component is called by the dial-up client (120) immediately before the dialing begins. The SDSetupDial component allows a user to terminate dialing, and provides the user information about the sending status updates and errors. In one embodiment of the present invention, information provided to a user is stored on a shared memory page.

[0045] In one or more embodiments of the dial-up client (120), using the above-mentioned components, automates the authentication process using a hidden terminal operating in terminal mode. Terminal mode allows data to be transferred via a traditional telephone line in text (ACSII) format. Additionally, the data may be encoded, *e.g.,* base 64 encoding. The dial-up client (120) provides an interface between the Security Management System with a client-side cryptographic function (128), *e.g.*, Entrust/Entelligence, and the Remote Access Switch (110). The client-side cryptographic function (128) provides an interface that can be used to access user certificates from a security device (106) via the appropriate

hardware and software components. The client-side cryptographic function (128) is also responsible for responding to a challenge, from the server-side cryptographic function (130) with a signed response string. The signing of the response string includes first hashing data to be sent using a hashing algorithm, such as MD5. The hashed data is then encrypted using a PKI encryption algorithm using the private key of the sender. The result is a digital signature of the response string.

[0046] Additionally, all data passed between the client computer (102) and the PKI-Bridge (124) is modified by the Custom Script DLL (122) to coincide with SDAP. Further, when sending the signed response string, the formatted signed response string is divided into packets and sent to the PKI-Bridge (124). This is required because response strings are typically large, *e.g.,* 3K bytes, thus it is difficult to send as one package due to bandwidth limitations of a data line, *e.g.,* telephone lines.

[0047] In one or more embodiments, the dial-up client (120) may perform several additional functions. The dial-up client (120) allows a dial-up user to dial into the Remote Access Switch (110) (*e.g.*, Shiva, Cisco, or other access vendor that support RADIUS servers) using a number selected by the user and provide status information about the dial-up connection. The dial-up client (120) also establishes PPP protocol and interfaces with e-mail service components for a phone book and associated protocols. For security purposes, the dial-up client (120) does not store any part of the information obtained from the client-side cryptographic function (128). Also, the dial-up client (120) does not store the challenge string or the signed response string.

[0048] In one or more embodiments, the dial-up client (120) provides several user interface components, including a phone number and modem setup screen, a connection information dialog box, a dial-up monitor, numerous error dialog